PURDUE UNIVERSITY

QUALITY PROGRAMMABLE VECTOR PROCESSORS FOR APPROXIMATE COMPUTING

Swagath Vekataramani[1], Vinay Chippa[1], Srimat Chakradhar[2], Kaushik Roy[1], Anand Raghunathan[1]

[1]Integrated Systems Laboratory
School of ECE, Purdue University
[2]NEC Laboratories America

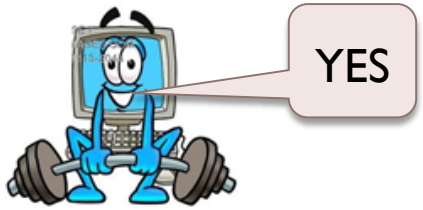
International Symposium on Microarchitecture 2013

COMPUTERS == PRECISE CALCULATORS

TASK

$\frac{433}{21} > 20.4$

```
float x = 433/21
float y = 20.4
(x > y) ? YES :NO
```

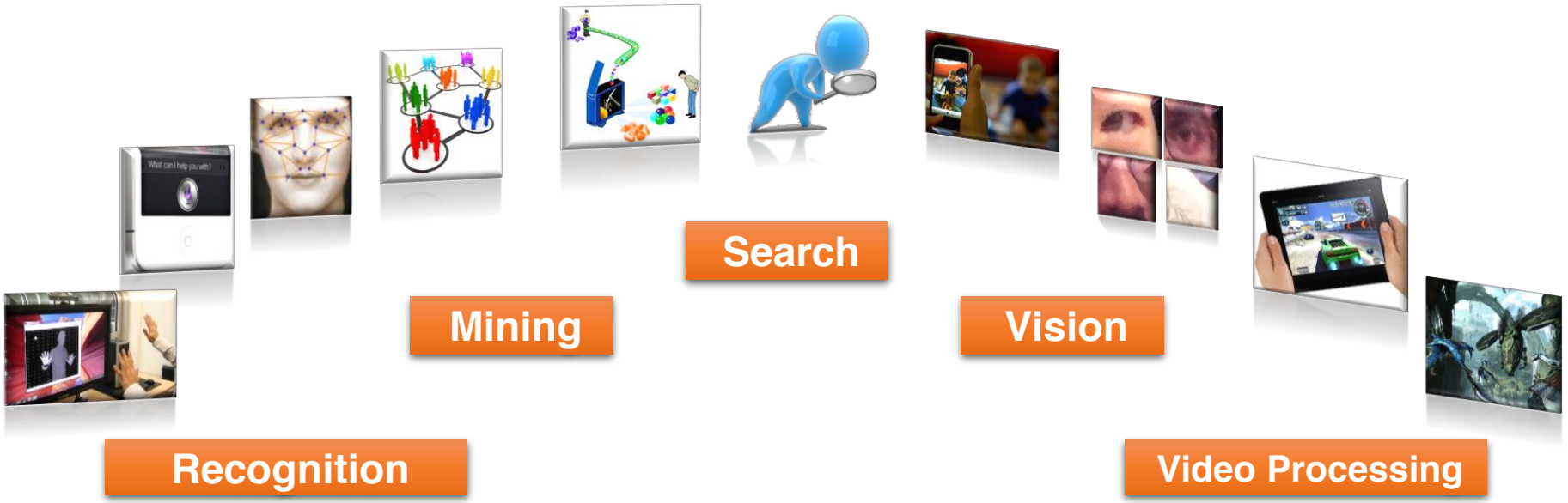


$\frac{433}{21} > 1$

```
float x = 433/21
float y = 1
(x > y) ? YES :NO
```

- Leads to inefficiency
- And, an overkill (for many applications)

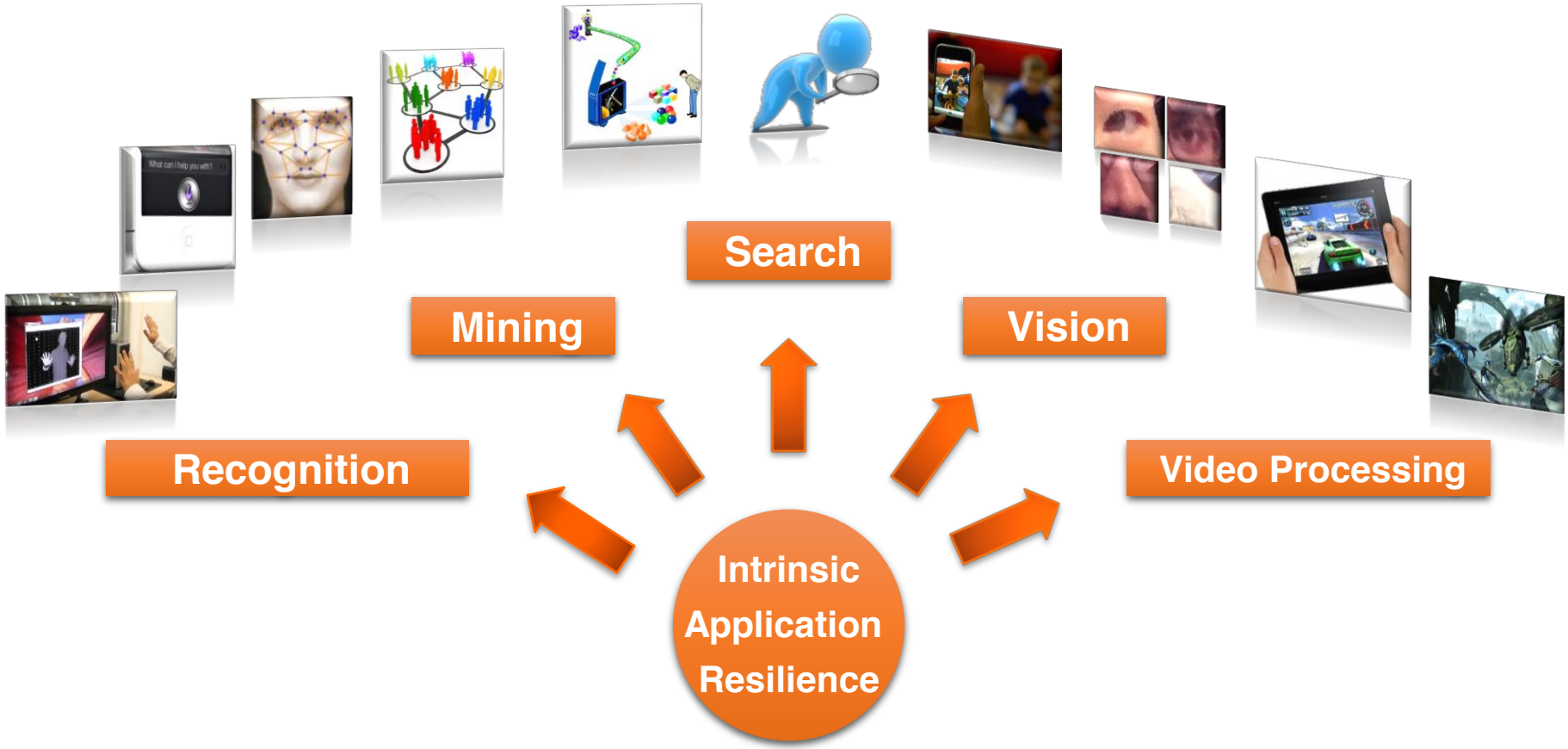
Evolving Application Landscape



- Relaxed notion of correctness
  - Results cannot be arbitrary either

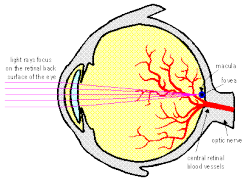
Good enough answers !!!

Intrinsic Application Resilience: A new dimension to optimize HW & SW



- The ability to produce outputs of acceptable quality despite many of their computations executed *imprecisely*

Intrinsic application resilience: Sources



Perceptual Limitations

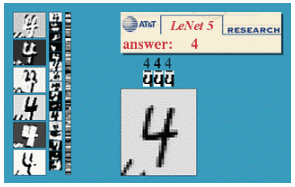
Redundant Input Data

Statistical Probabilistic Computations

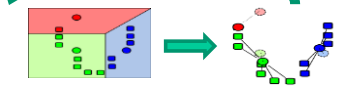
'Noisy' Real World Inputs

Intrinsic Application Resilience

Self-Healing



Repeat until convergence

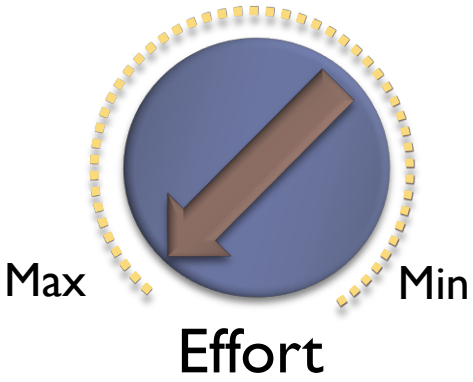


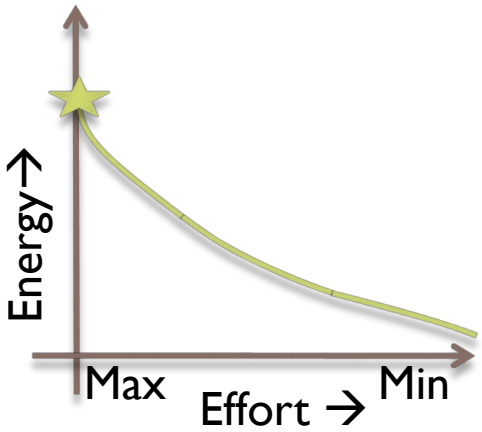
Compute distances & assign points to clusters

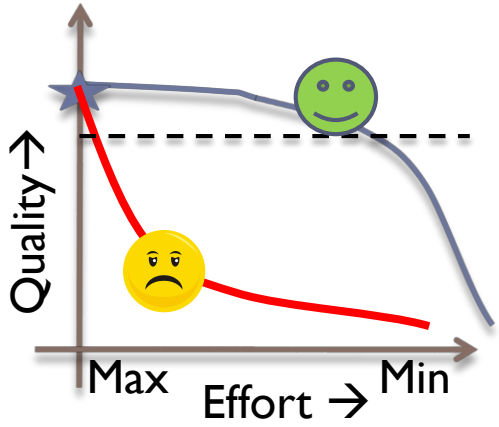
Update cluster means

Approximate Computing: Design Philosophy

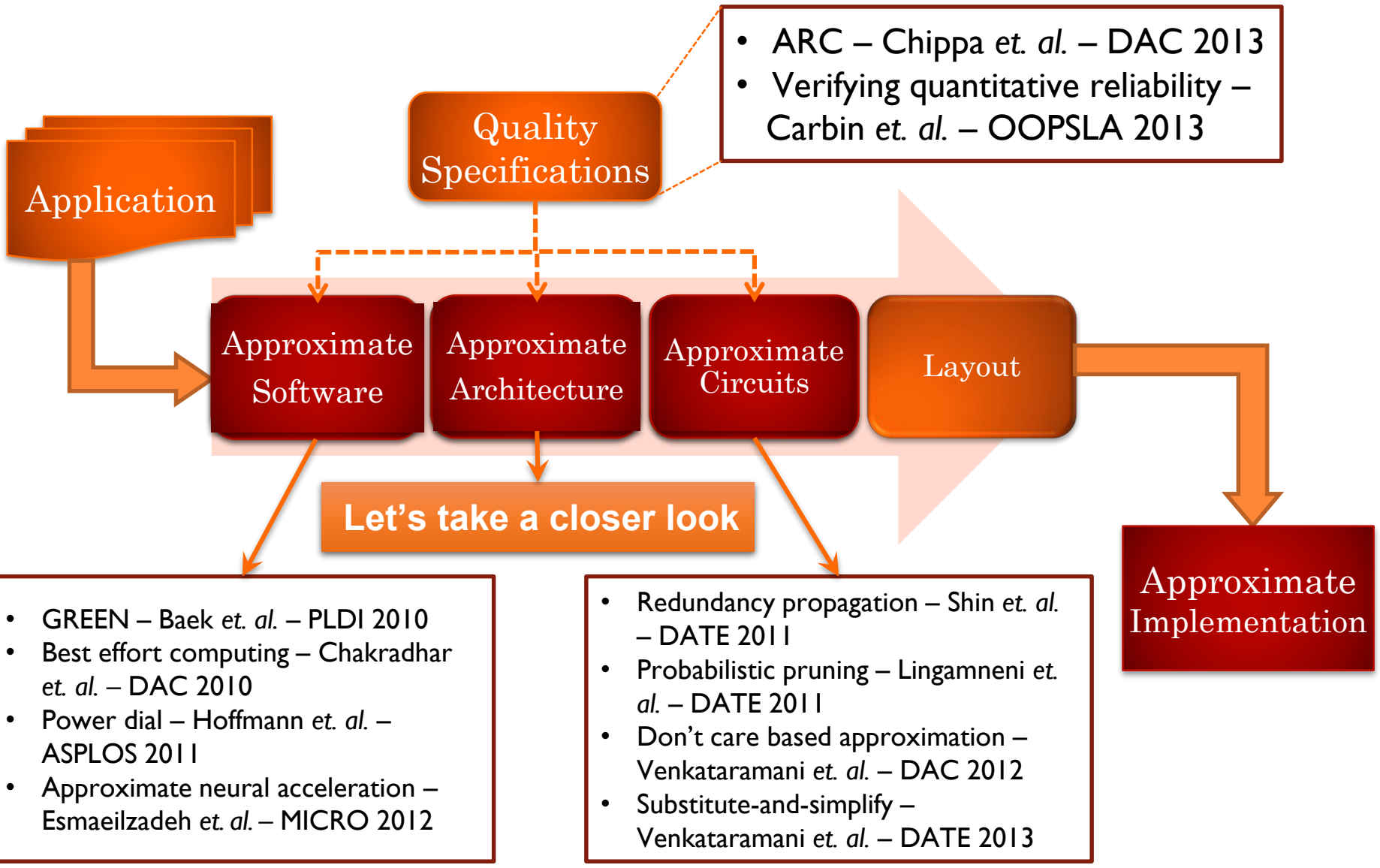
- Systems that can modulate the effort expended towards quality of results
  - Higher effort → Higher quality but higher energy
- How do we get the best Q vs. E tradeoff?
  - Disproportionate benefit







Approximate Computing Design Techniques: Overview



Approximate Architecture

Algorithm-specific accelerators

Domain-specific accelerators – image, video …

Programmable accelerators (GPGPUs, MIC) / Vector processors

General purpose processors/ Multicores

- ANT – Hedge *et. al.* – ISLPED 1999
- Significance driven computing – Mohapatra et. al. – ISLPED 2009
- Scalable effort hardware – Chippa et. al. – DAC 2010, DAC 2011

Application specific designs

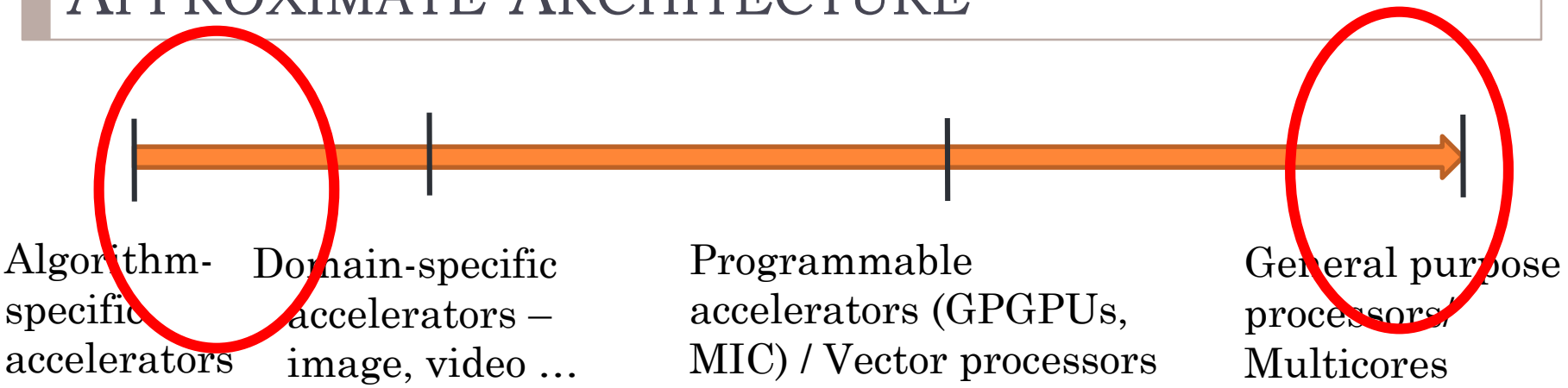
- ERSA – Leem *et. al.* – DATE 2010
- Stochastic processor – Narayanan *et. al.* – DATE 2010

Cores of different reliabilities

- Truffle – Esmaeilzadeh *et. al.* – ASPLOS 2012
- EnerJ – Sampson *et. al.* – PLDI 2011

Accurate and approximate instructions

Approximate Architecture



Pros:

☺ Large energy benefits

☺ Broader applicability

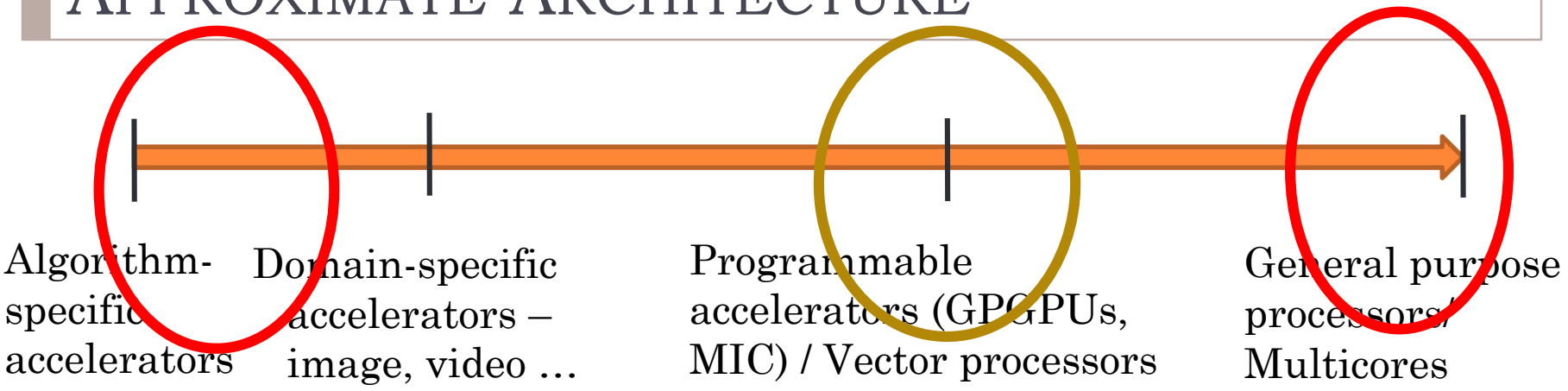
Challenges:

☹ Limited applicability

☹ Inherently limited energy benefit – Dominated by control front-ends that cannot be approximated

☹ Allow arbitrary errors in hardware – limits the fraction of computations that can be approximated

Approximate Architecture



Opportunity:

☺ Wide range of applications – fine grained parallelism

☺ SIMD: Control overheads amortized over many execution units

☺ Need quality guarantees from HW – We will address that!

CONTRIBUTIONS

- **Quality programmable processors**

  An abstract model for programmable approximate processors

- **QUORA**

  A quality programmable 1D/2D vector processor

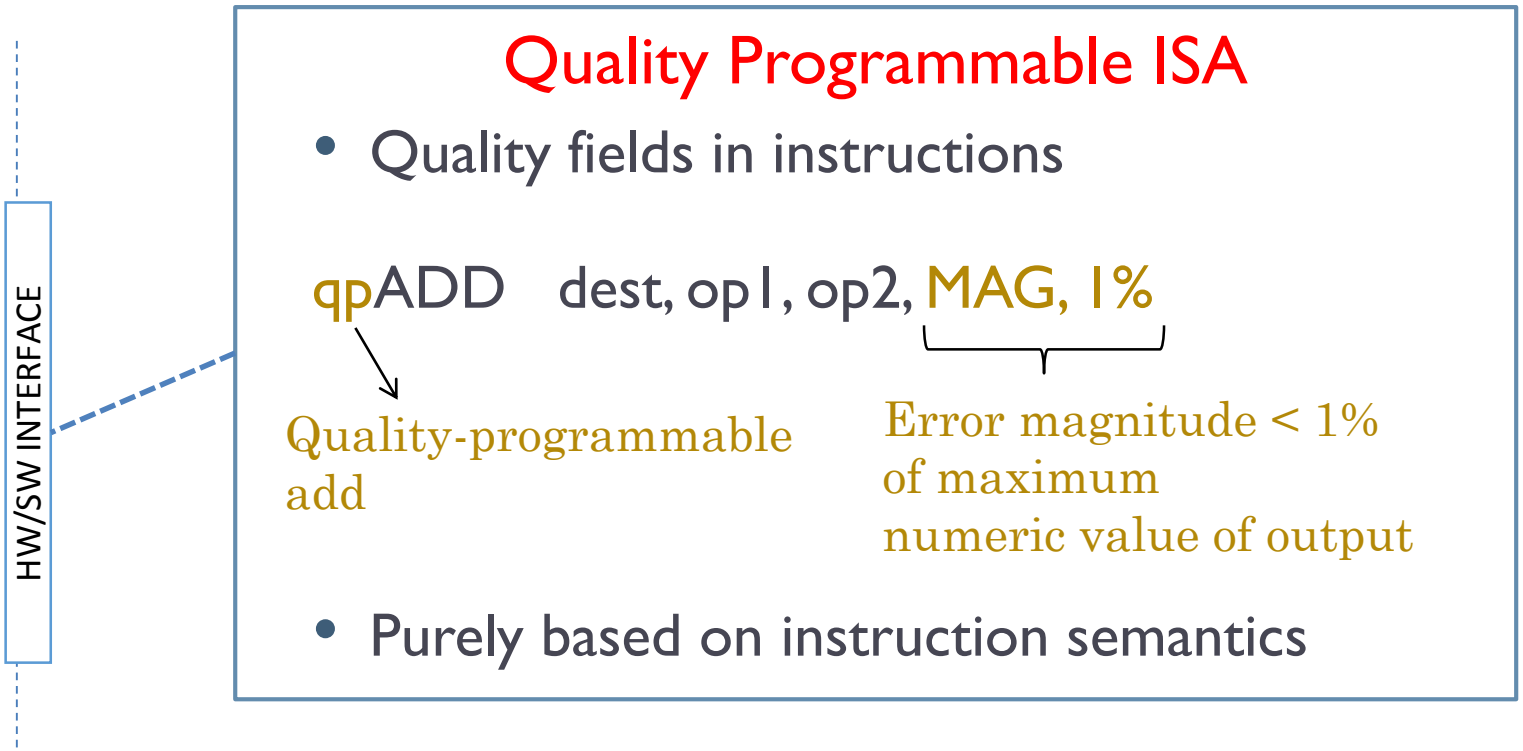
Quality programmable processors

Requirements:

- HW/SW interface for applications to expose resilience
- Micro-architecture that can translate resilience to efficiency

Quality Programmable Processors

- Quality Programmability: Ability to specify beyond what can be accurate & approximate to HW
- Notion of *quality* explicitly built into the instruction set



Need For Quality Programmability

- Errors injected randomly in x86 instructions – arbitrary vs. bounded

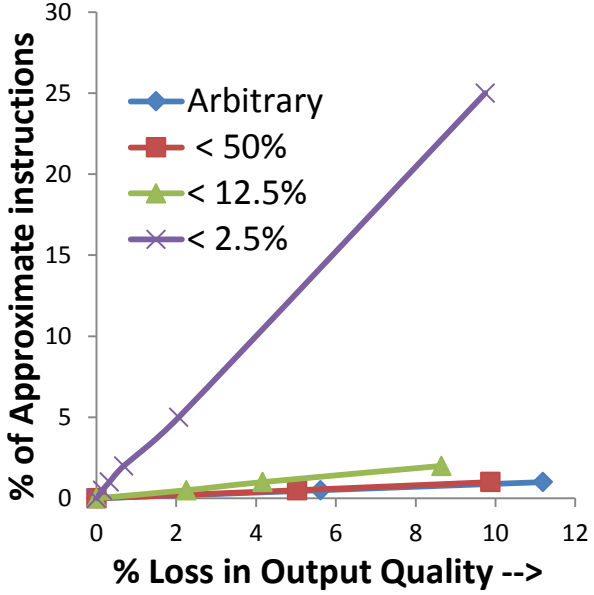
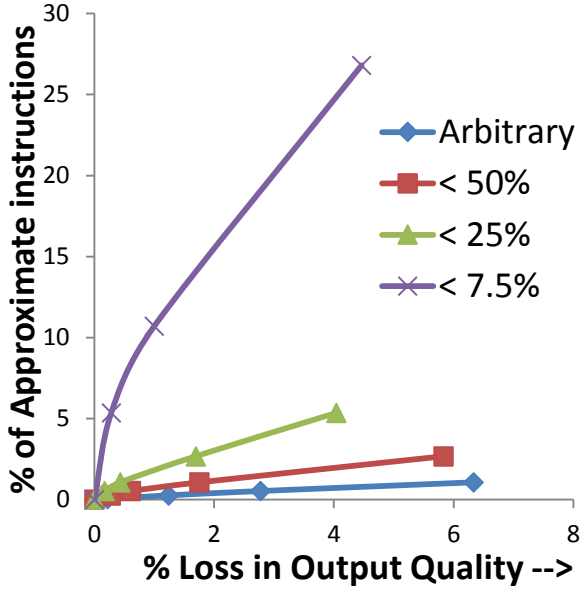


Image Segmentation (K-means)



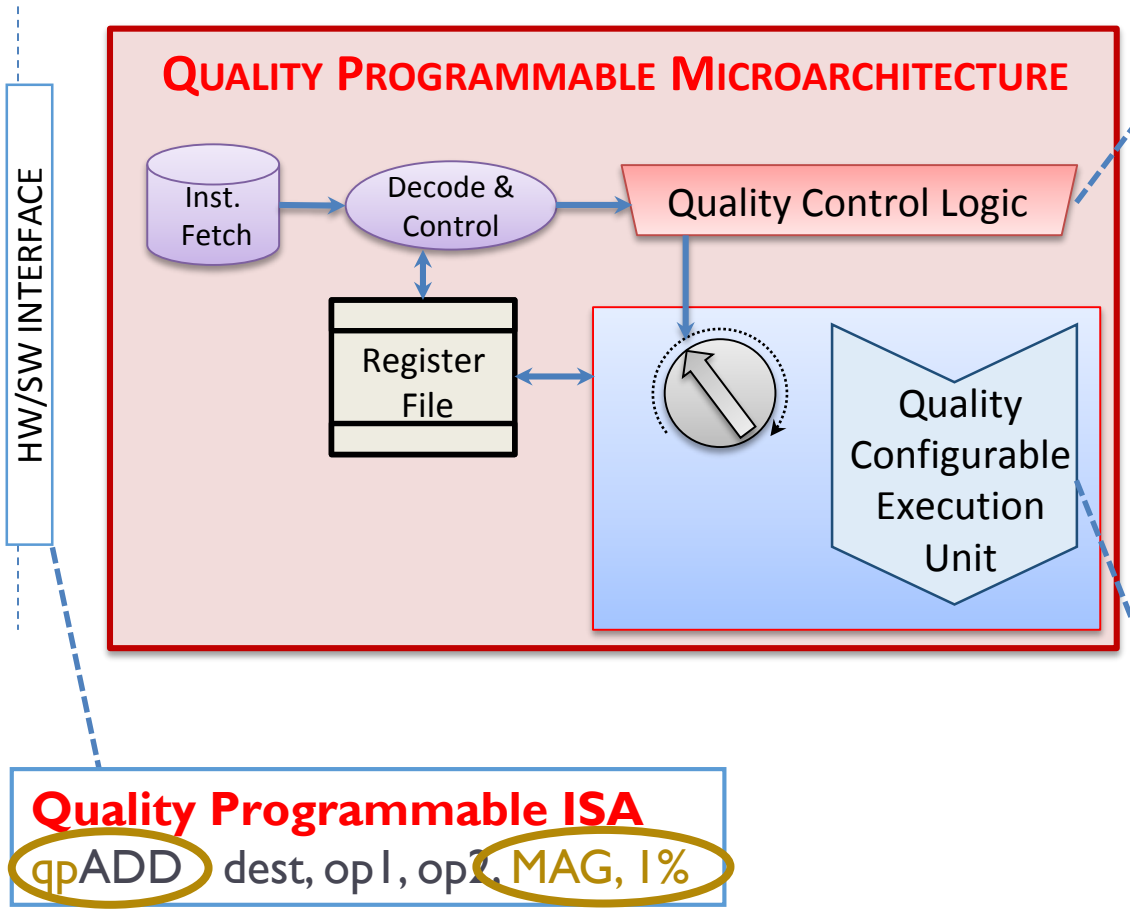
Handwritten Digit Recognition (SVM)

25-100X improvement in number of approximate instructions

Constraining errors ➔ much greater opportunity to approximate!!!

Quality Programmable Micro-architecture

- Micro-architecture *guarantees* instruction-level quality

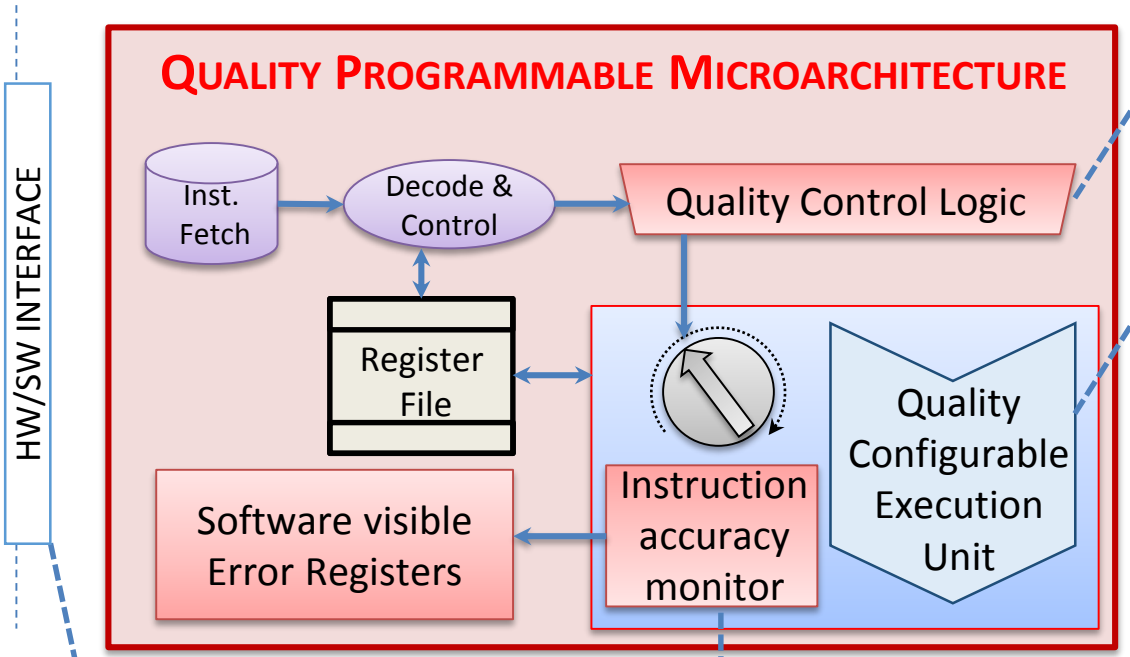


- Translate instruction quality specification into accuracy knobs built in hardware

- Capable of executing instructions with different quality levels
- Any approximate HW design technique
  - e.g. precision scaling

Quality Monitors and Error Feedback

- Micro-architecture provides *feedback* on instruction-level quality to software



Translate instruction quality → HW accuracy knobs

Capable of executing instructions with different quality levels

- uArch may not be able to use all the flexibility
  - Conservative quality translation
  - Error variance over i/p data
- Estimate actual error

Quality Programmable ISA

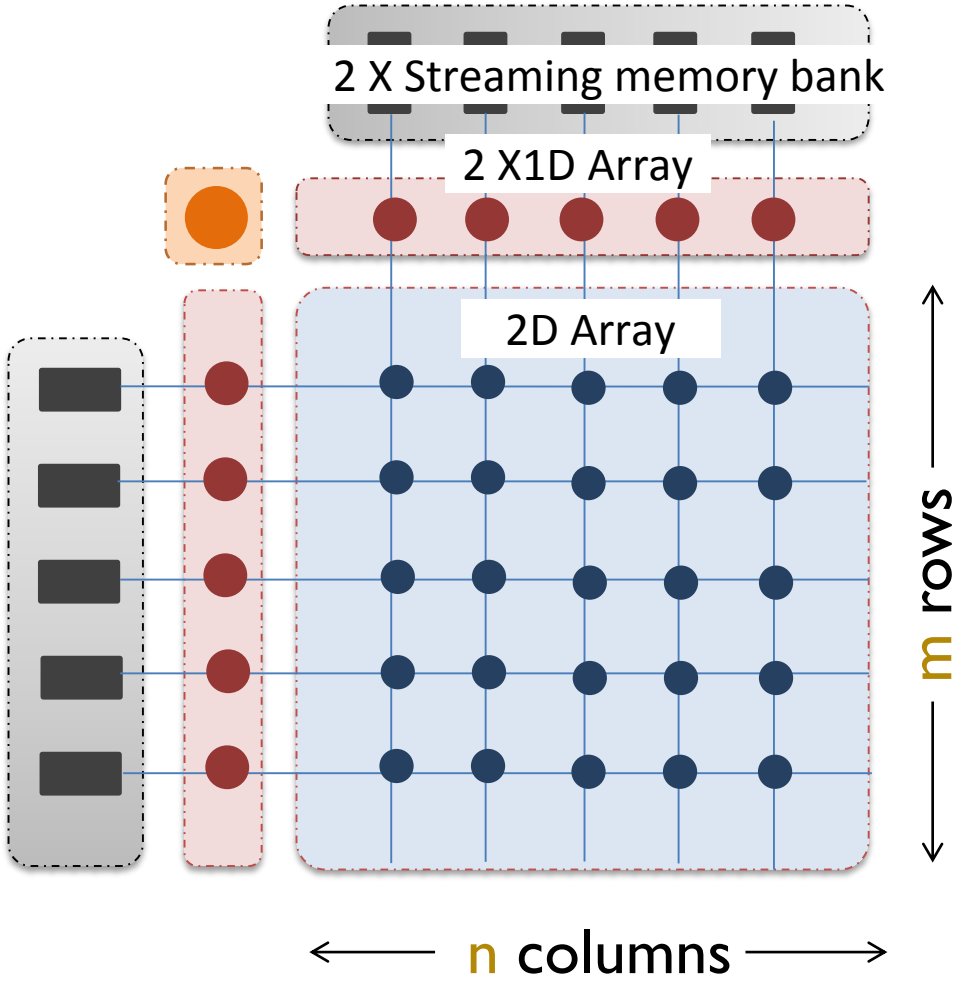
qpADD dest, op1, op2, MAG, 1%

QUORA

Quality programmable 1D/2D vector processor

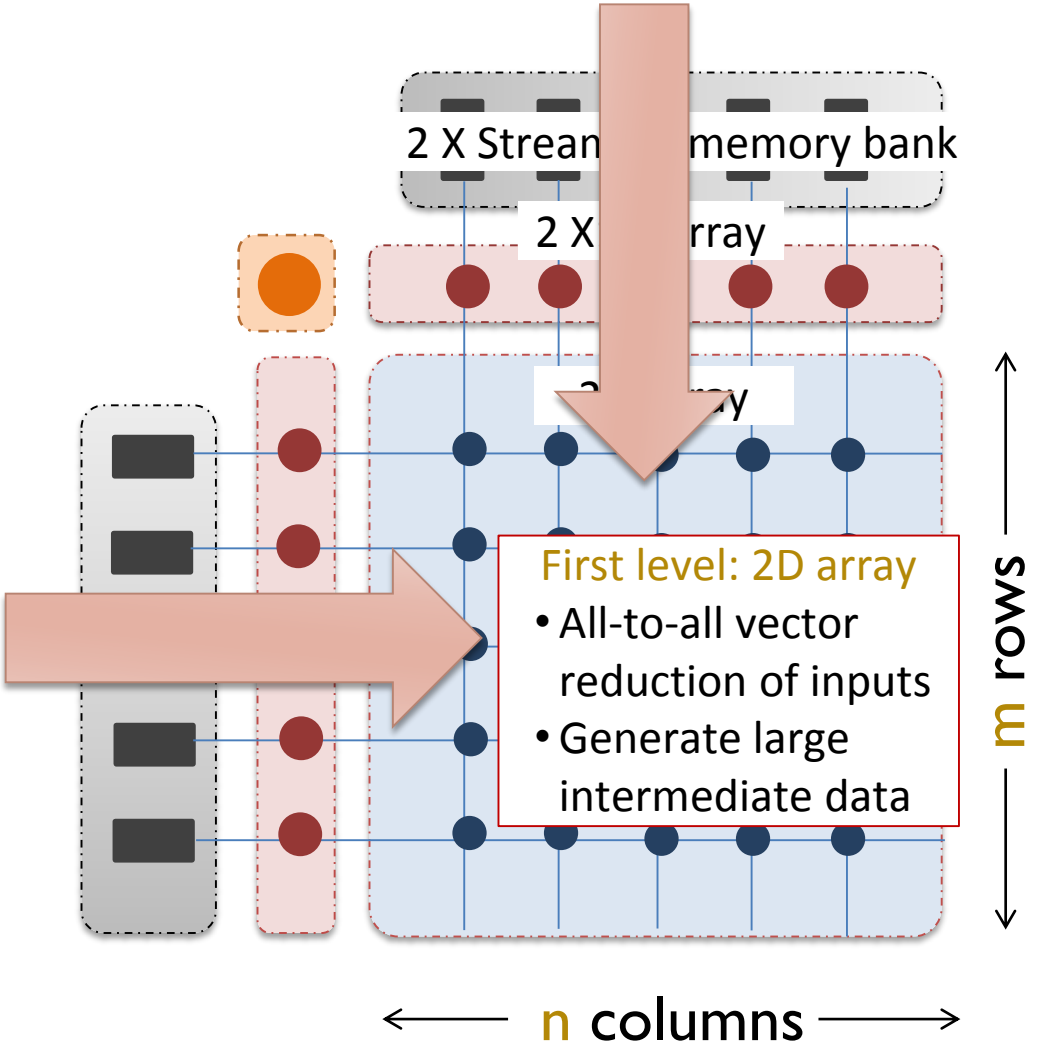
QUORA: OVERVIEW

- 3-tier processing element hierarchy
  - 2D array PEs
  - 2 sets of 1D array PEs
  - One scalar PE
- 2 streaming memory banks along the array borders



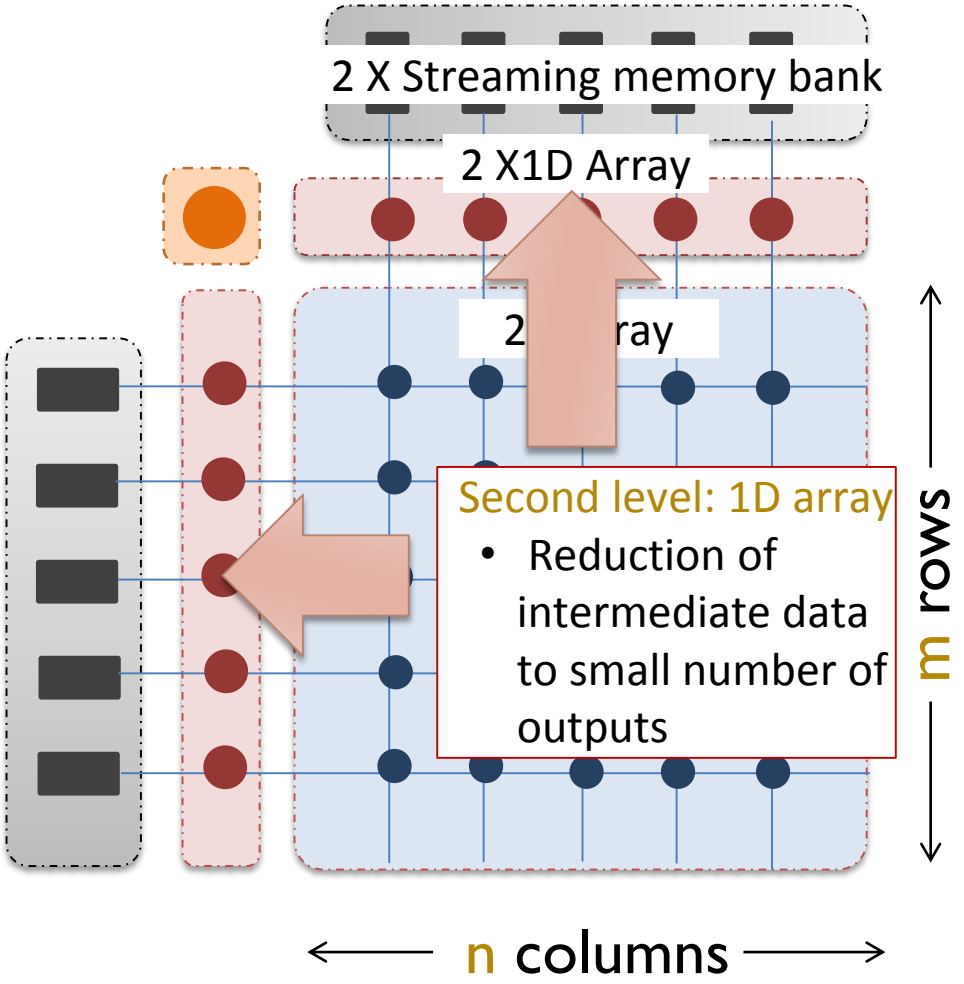
QUORA: OVERVIEW

- 3-tier processing element hierarchy
  - 2D array PEs
  - 2 sets of 1D array PEs
  - One scalar PE
- 2 streaming memory banks along the array borders
- Application characteristic: 2 levels of reduction operations

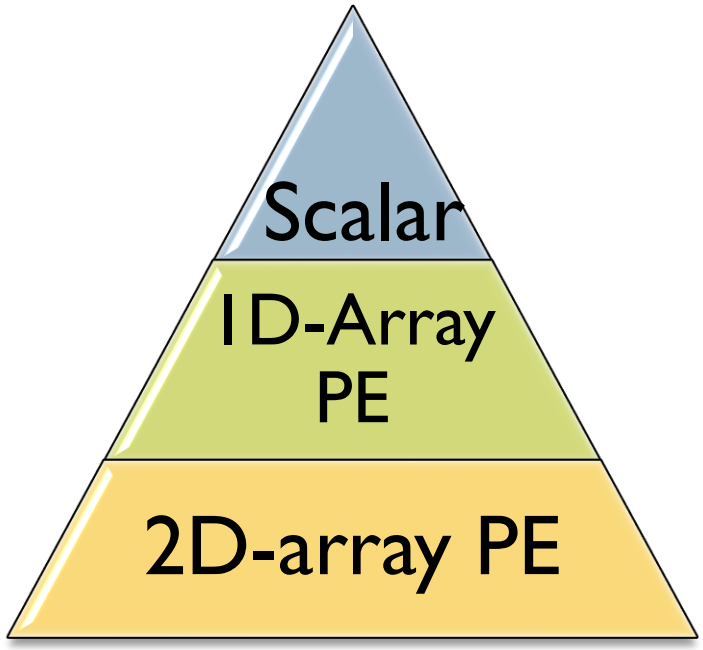


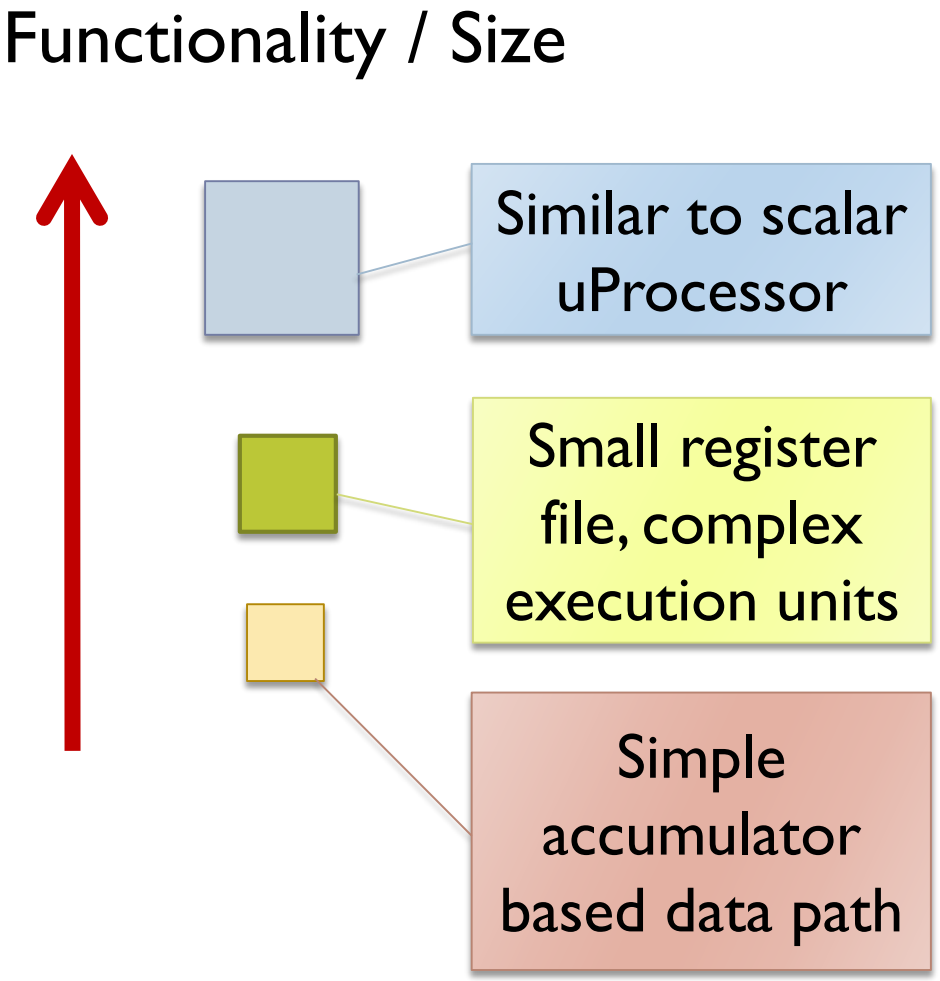
QUORA: OVERVIEW

- 3-tier processing element hierarchy
  - 2D array PEs
  - 2 sets of 1D array PEs
  - One scalar PE
- 2 streaming memory banks along the array borders
- Application characteristic: 2 levels of reduction operations

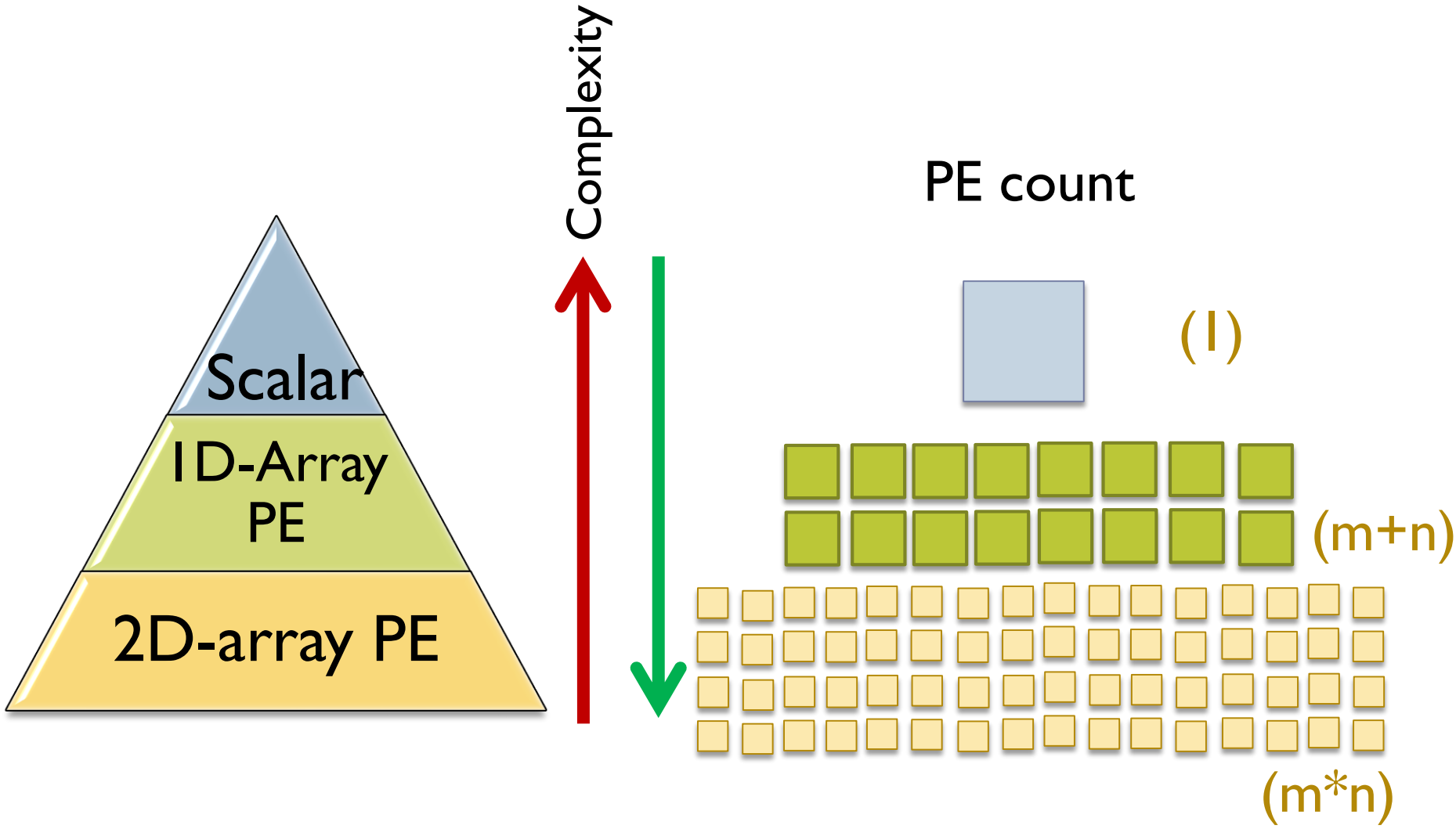


Processing Element Hierarchy

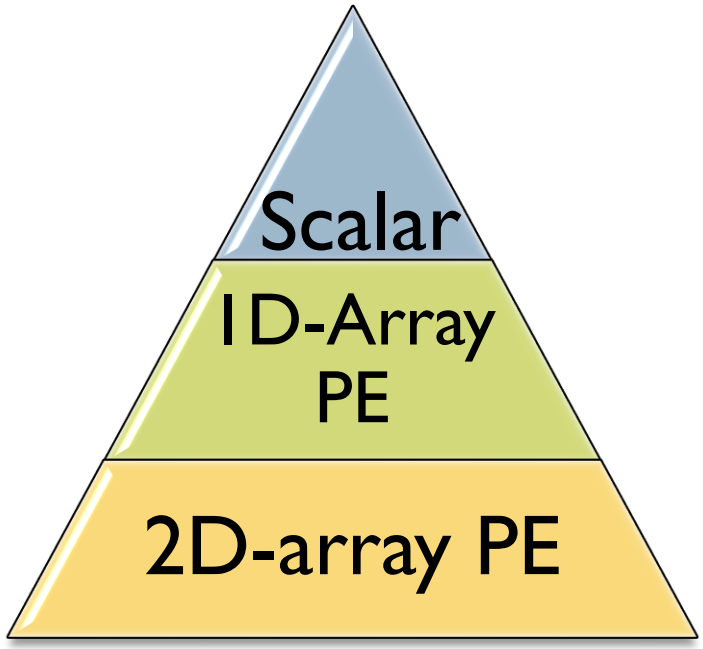


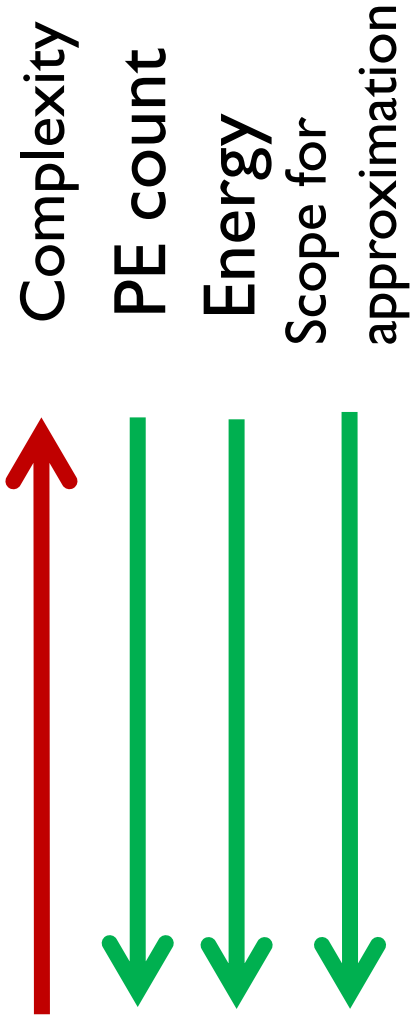


Processing Element Hierarchy

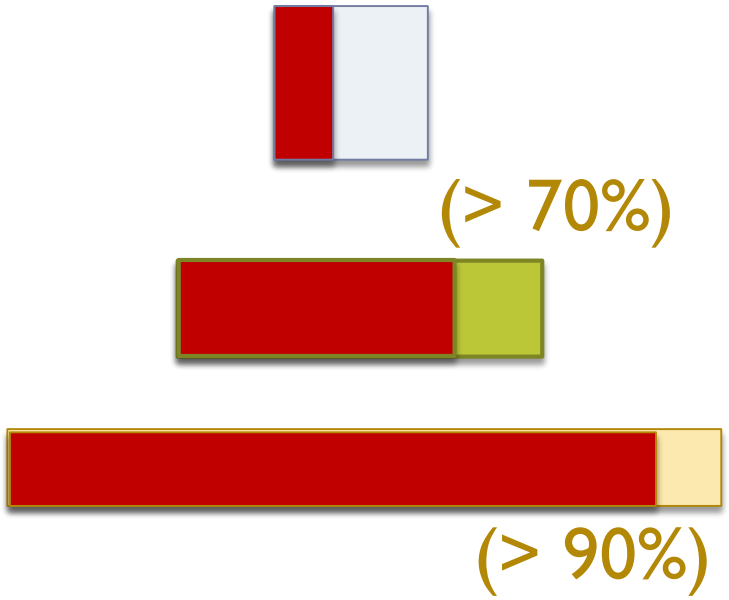


Processing Element Hierarchy

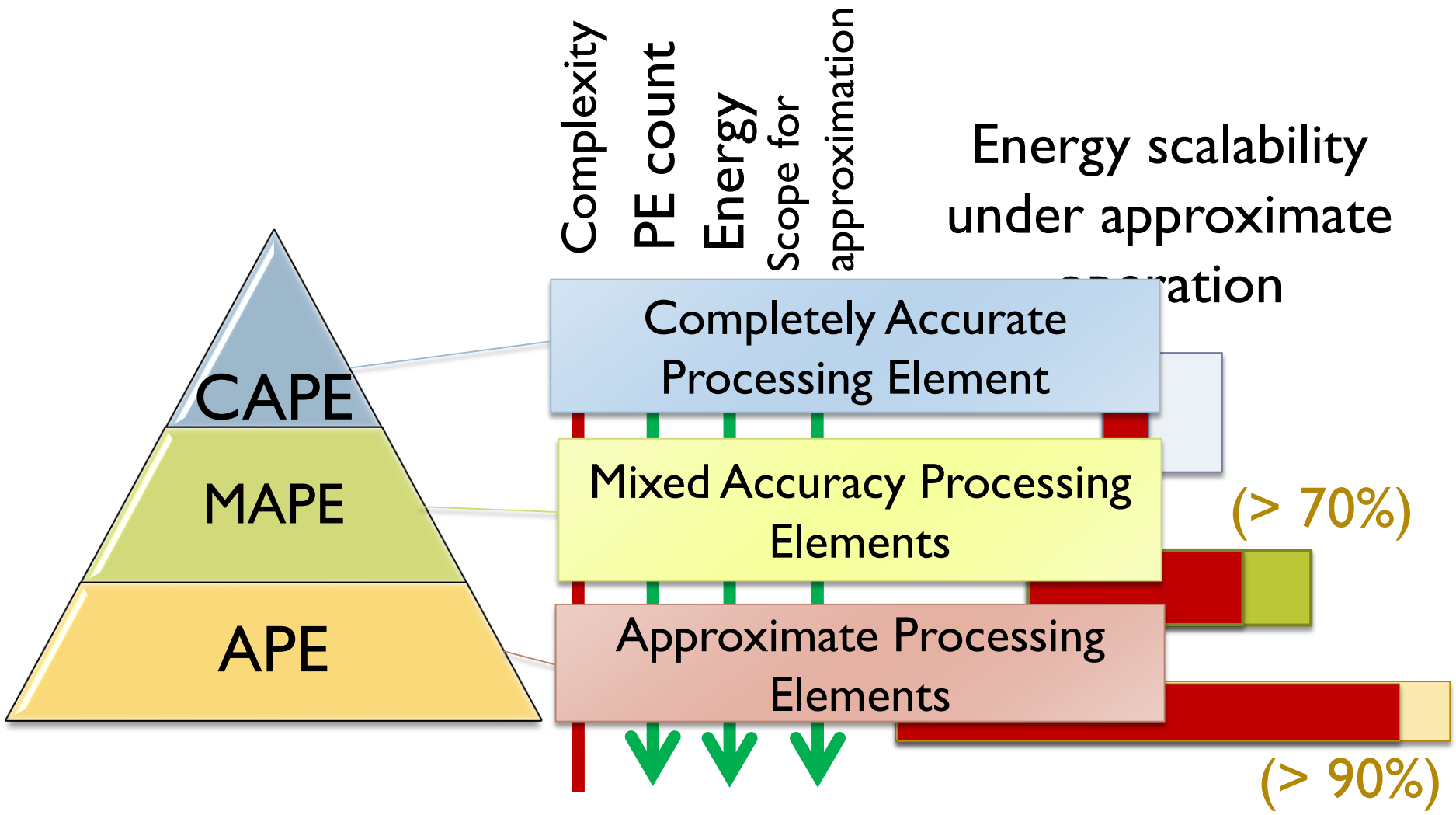




Energy scalability under approximate operation



Processing Element Hierarchy



3-tiered PE hierarchy enables larger energy benefits from approximate computing (while matching application characteristics)

QUORA: INSTRUCTION SET ARCHITECTURE

- 47 Instructions – 9 APE, 22 MAPE, 13 CAPE, 3 SM

Inst. Type	Instruction
Scalar Instructions	LDRI Rd, value
	ADDR Rd,Rs1,Rs2
	BEZ Rs, Rel. address
	HALT
Streaming Memory instructions	LDSM R_length, stride, burst, R_st_add
2D Array Instructions	**qpMAC** R_length, R_row_enb, R_col_enb, ***R_q_type, R_q_amt***
	qpMOD2 R_length, R_row_enb, R_col_enb, ***R_q_type, R_q_amt***
	STR <r/c>, R_stride, R_burst, R_st_add, R_row_enb, R_col_enb

Inst. Type	Instruction
1D Array Reduction Instructions	**qpACC** <r/c>, R_row_enb, R_col_enb, ***R_q_type, R_q_amt***
	qpMIN <r/c>, R_row_enb, R_col_enb, ***R_q_type, R_q_amt***
1D Array Streaming Instructions	SEQ R_length, SReg, R_row_enb, R_col_enb
1D Array Self-Operand Instructions	MVASR <r/c>, R_<r/c>_enb, SReg
	qpADDX <r/c>, R_<r/c>_enb, Sreg, ***R_q_type, R_q_amt***
	qpMUL <r/c>, R_<r/c>_enb, Sreg, **R_q_type, R_q_amt**
	STMCG <r/c>, R_<r/c>_enb, SReg

QUORA – QUALITY PROGRAMMABLE INSTRUCTIONS

- APE and MAPE instructions extended with 2 additional quality fields

e.g. **qpMAC** R_length, R_row_enb, R_col_enb, ***R_q_type, R_q_amt***

- Type of error – 3 quality metrics

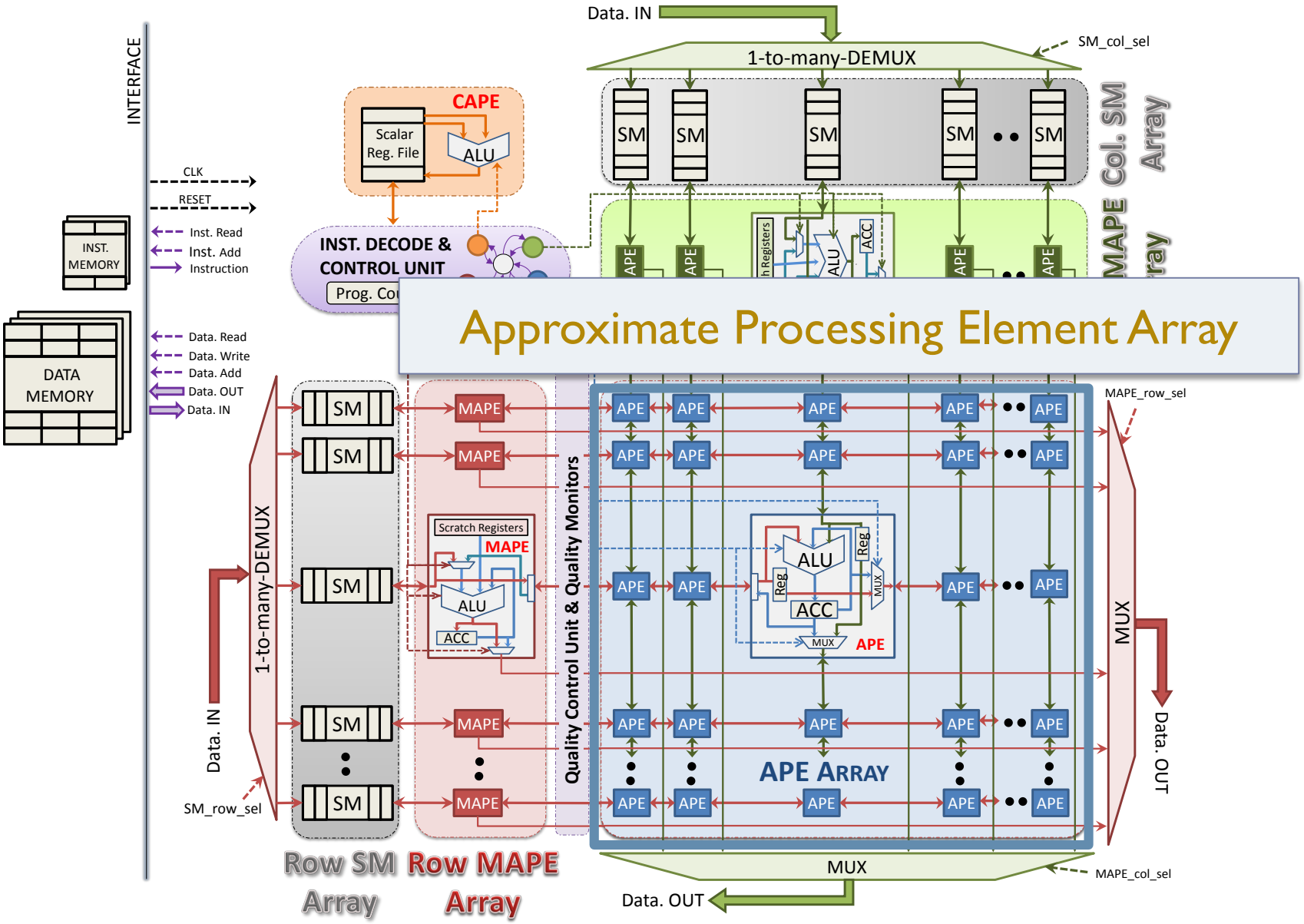
$$MaxErr = |Oorig - Oap_{prox}|$$

$$AveErr = \frac{\sum_{\forall inp} |O_{orig} - Oapp_{rox}|}{Total\ no.of\ inputs}$$

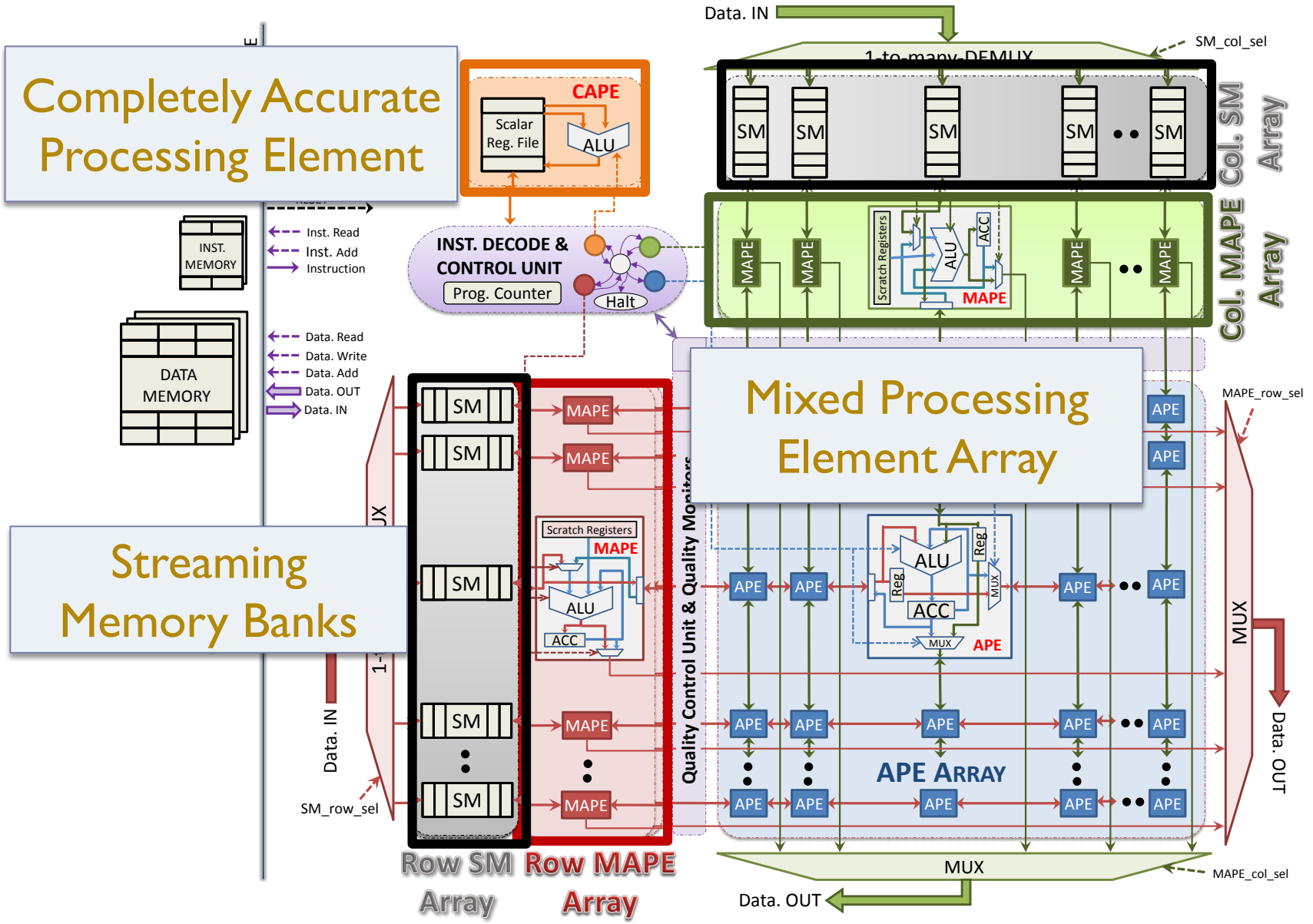
$$ErrProb = \frac{No.of\ inputs\ for\ which\ O_{orig} \neq O_{approx}}{Total\ no.of\ inputs}$$

- Amount of error

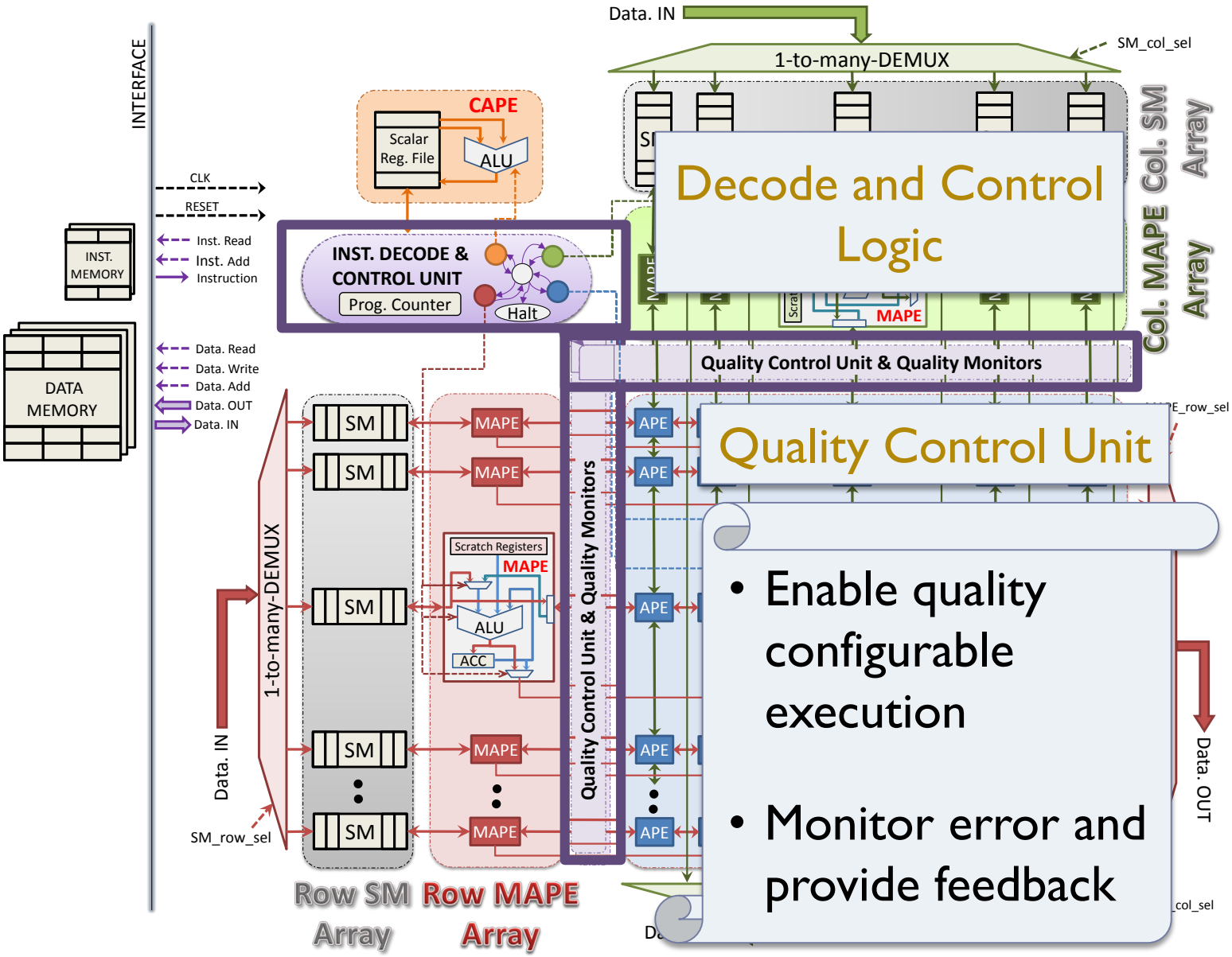
QUORA: QP 1D/2D VECTOR PROCESSOR



Quora: QP 1D/2D Vector Processor



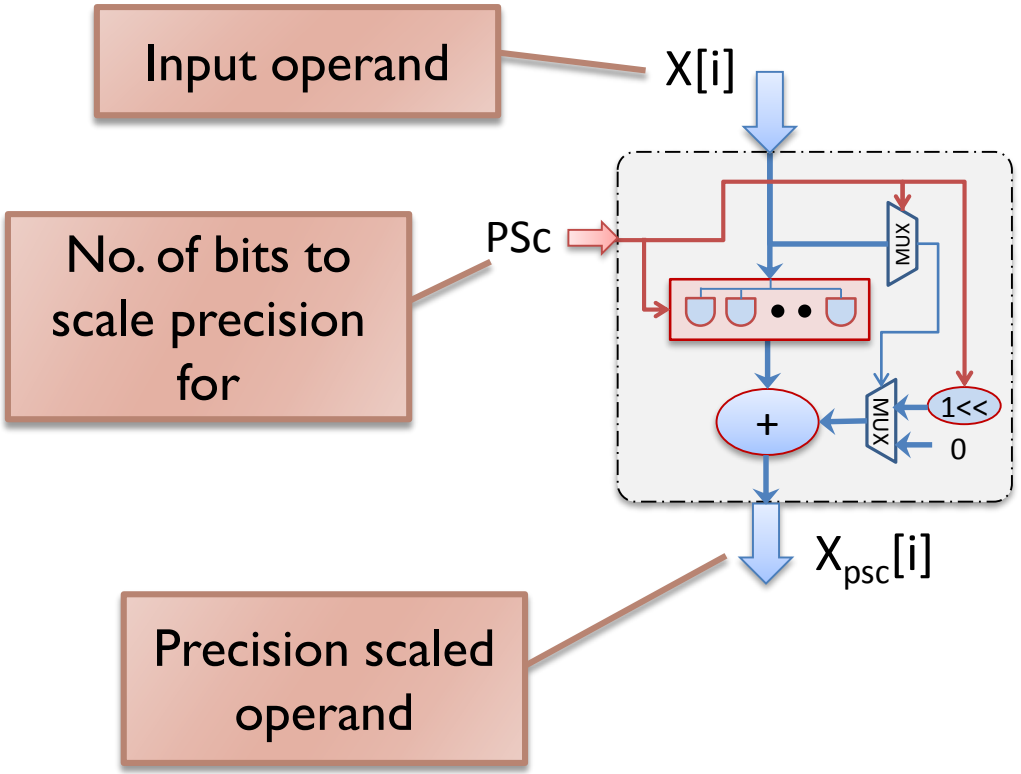
QUORA: QP 1D/2D VECTOR PROCESSOR



Quality Configurable Execution: Precision Scaling

- Scale the precision of input operands to APEs/MAPEs
  - 4 different flavors

- Up/down operand round-off



Up/Down Precision Scaling

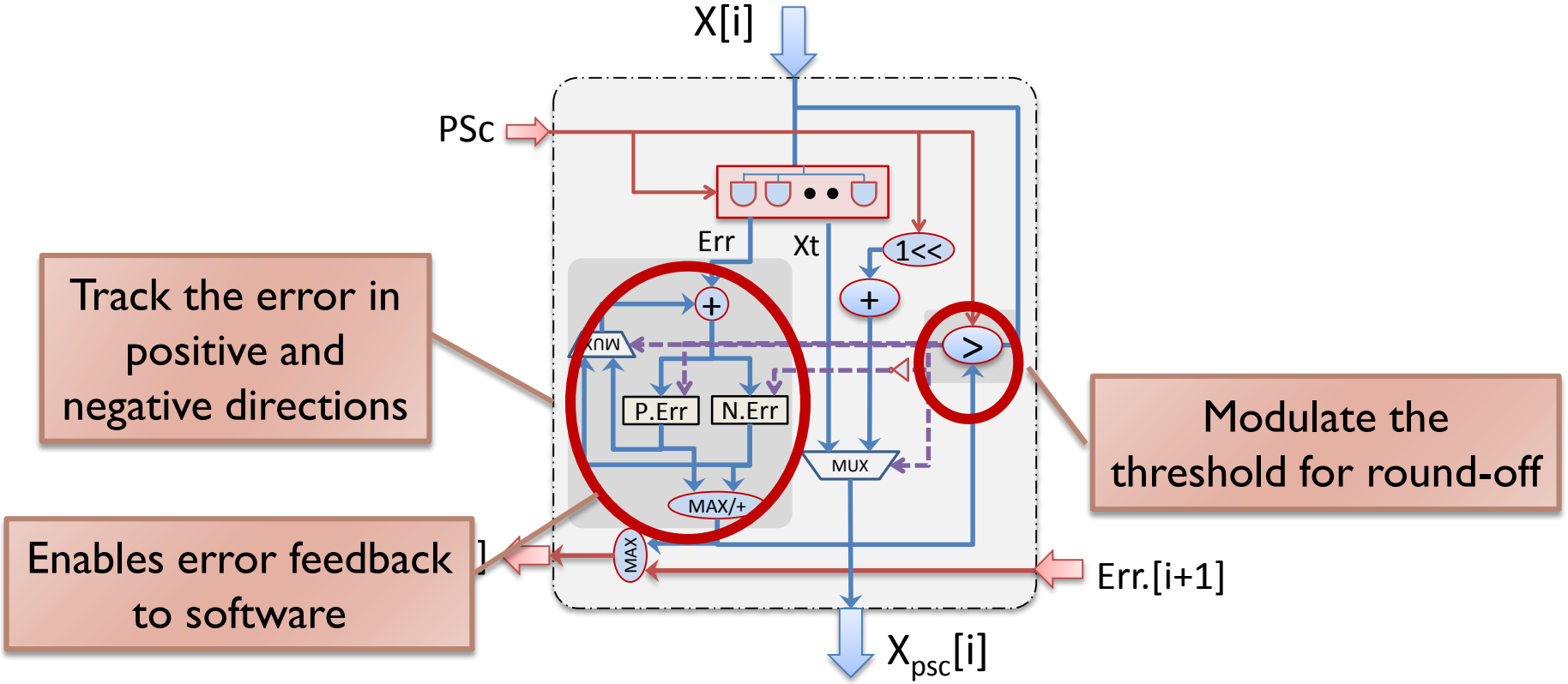
e.g., PSc == 3

X[2:0] >= 4 ?

Round up : Round down

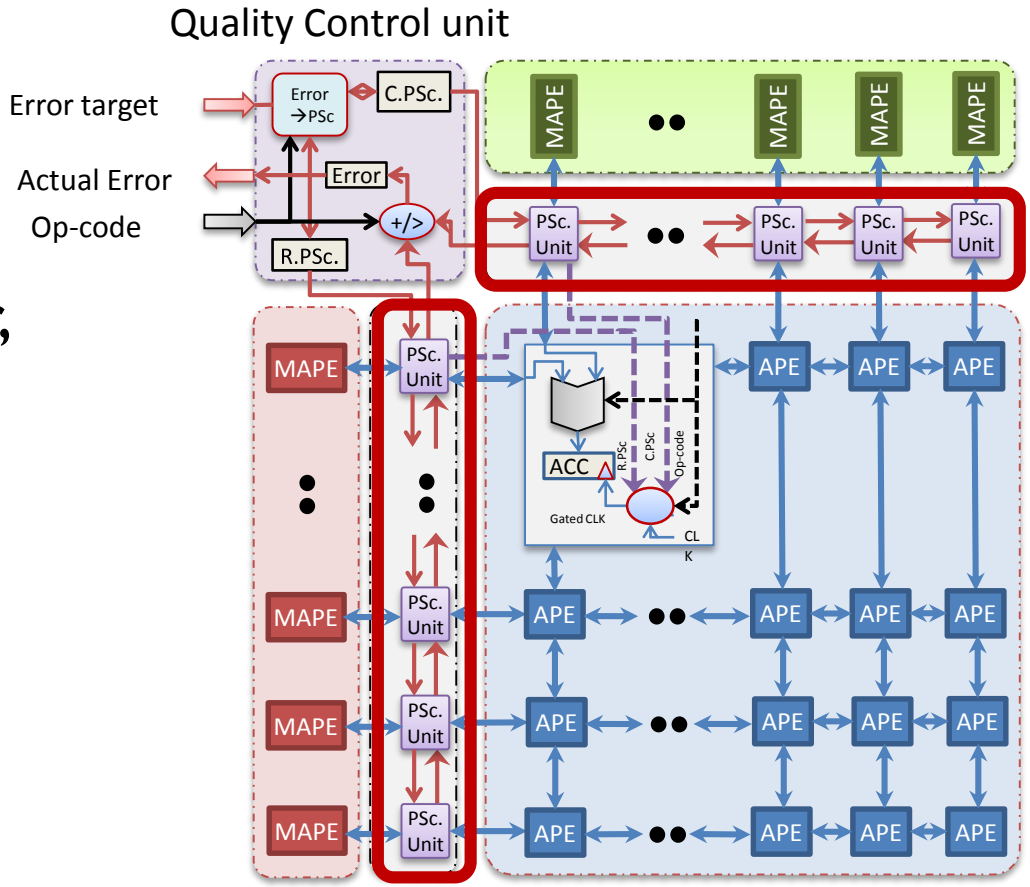
Precision Scaling with Error Compensation

- Quality specified at the output of vector operations
- Key idea: Compensate errors across many scalar operations to reduce overall instruction level error



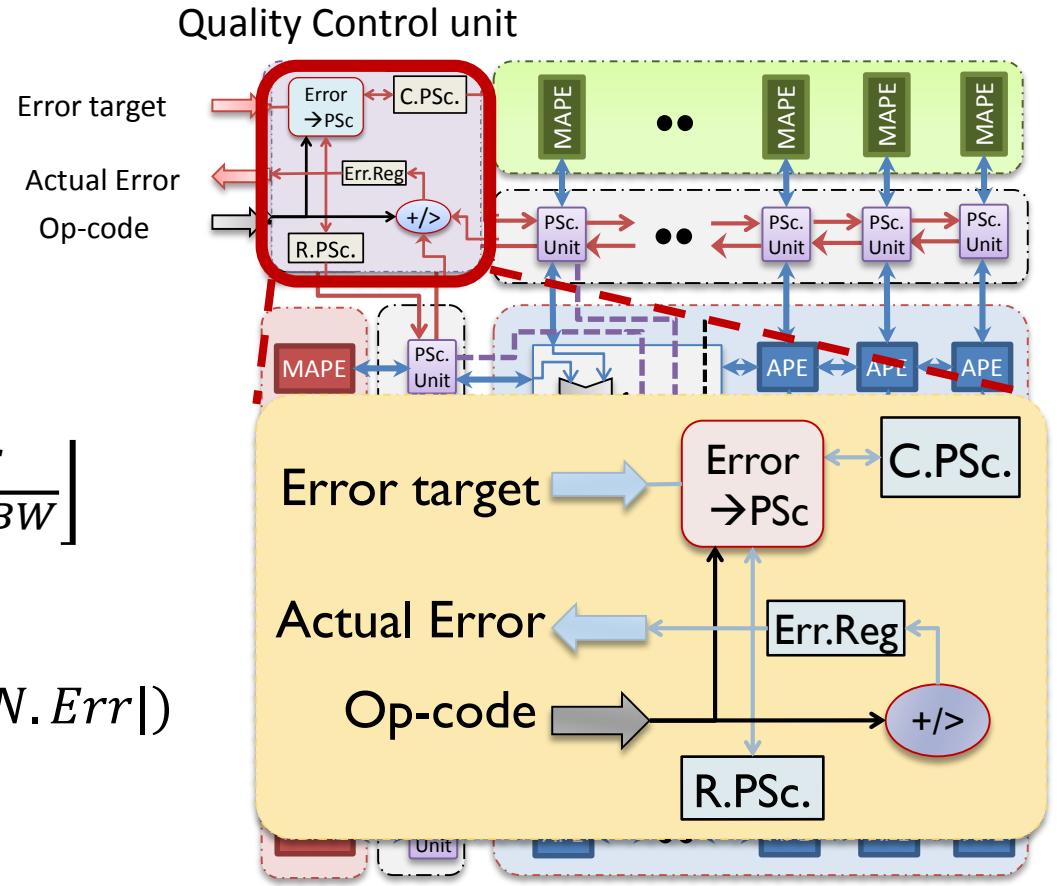
Precision Scaling: Array level view

- PSc. units located along the borders of the 2D array
- Benefits: Dynamic power, clock gating, precision of future instructions
- Overheads: PSc. units shared across PEs in a row/column. <1% of overall energy



Quality Control Unit

- Set PSc based on required instruction level quality bound



- *E.g.,* MAC operation

$$\textbf{C.PSc} = 1 + lg_2 \left\lfloor \frac{MaxErr}{length * 2^{BW}} \right\rfloor$$

$$\textbf{R.PSc} = 0$$

$$\textbf{E. Reg} = 2^{BW} * \max\left(|P.Err|, |N.Err|\right)$$

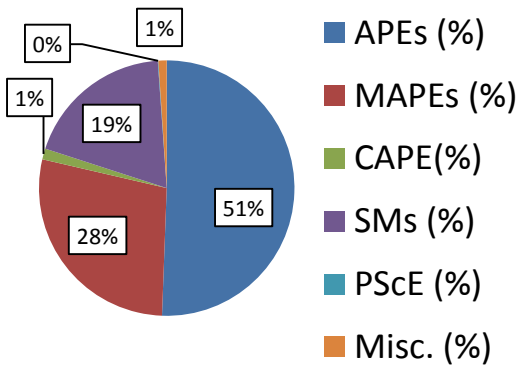
- More details in paper

Experimental Methodology

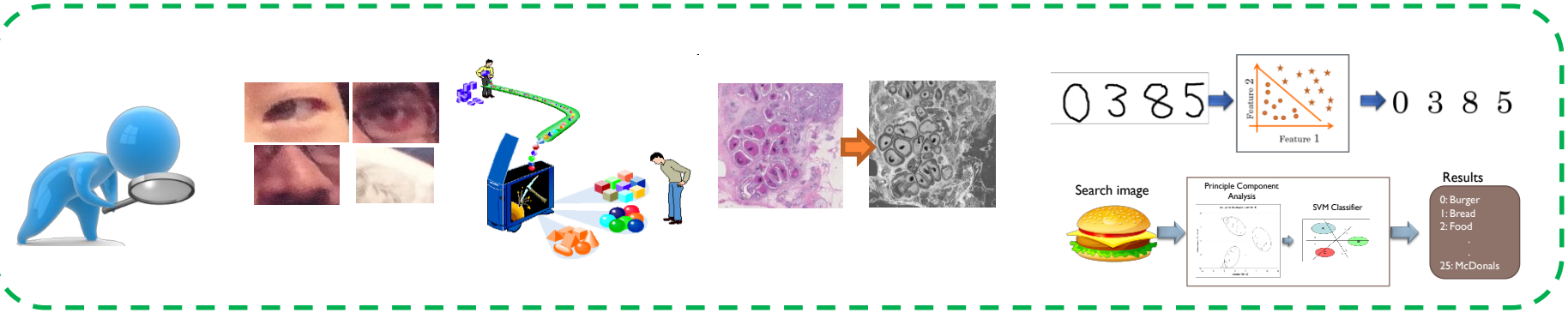
- RTL implementation of QUORA using Verilog HDL
- Synthesized to IBM 45nm technology node
  - Design flow: *Synopsys Design Compiler, ModelSim, Synopsys Power Compiler*

Micro-architectural Parameters	Value
Array Dimensions	16 X 16
No. of PEs (APEs + MAPEs+ CAPE)	289 (256 + 32 + 1)
Size of Register File – CAPE / MAPE	32 / 8
No. of SM elements	32
Depth of SM elements	64
Operating Frequency	250 MHz

Metric	Value
Feature Size	45nm
Area	2.6 mm^2
Power	367.8 mW
Gate Count	502042

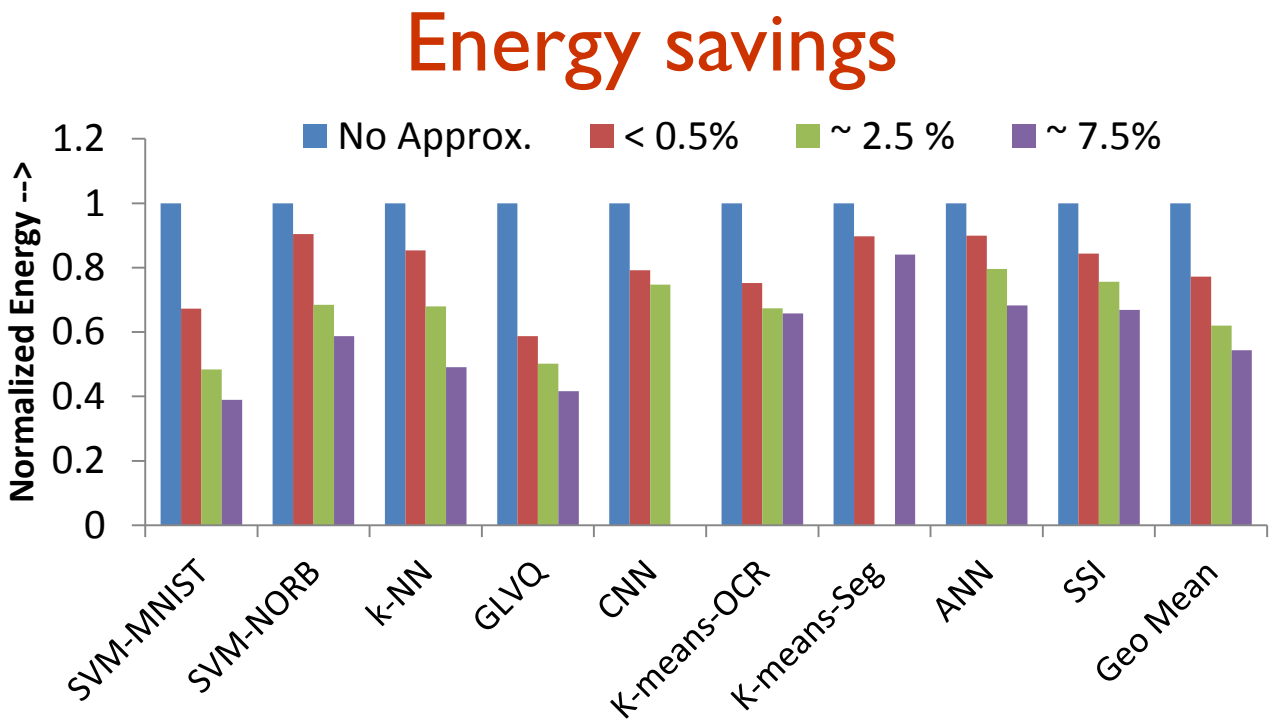


BENCHMARKS



Applications	Algorithm	Dataset	Quality Metric
Handwritten Digit Recognition (SVM-MNIST)	SVM	MNIST	Percentage classification accuracy
Object Recognition (SVM-NORB)	SVM	NORB	
Digit Classification (CNN)	CNN	MNIST	
Eye Detection (GLVQ)	GLVQ	Image set from NEC labs.	
Optical Character Recognition (k-NN)	K-NN	OCR digits	
Census Data Analysis (ANN)	ANN	Adult	
Document Search (SSI)	SSI	Subset of Wikipedia	No. correct in top 25 results
Image Segmentation (K-Means-Seg)	K-means	Berkeley dataset	Mean distance of clustered points from respective centroids
Optical Character Clustering (K-Means-OCR)	K-means	OCR digits	

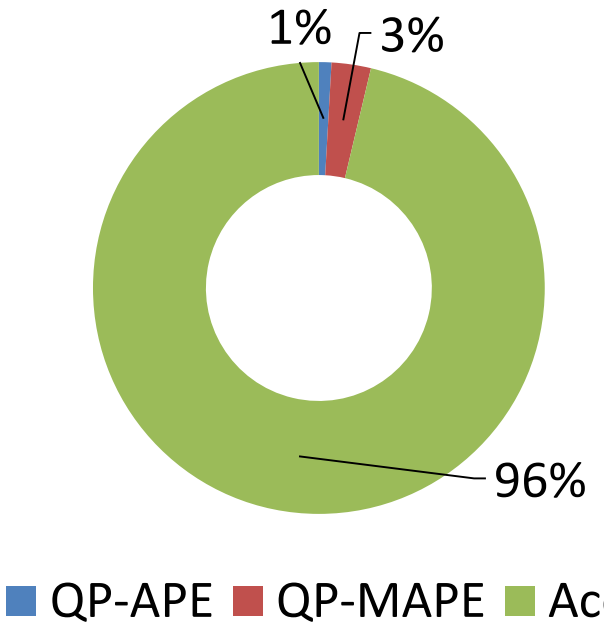
Results Summary



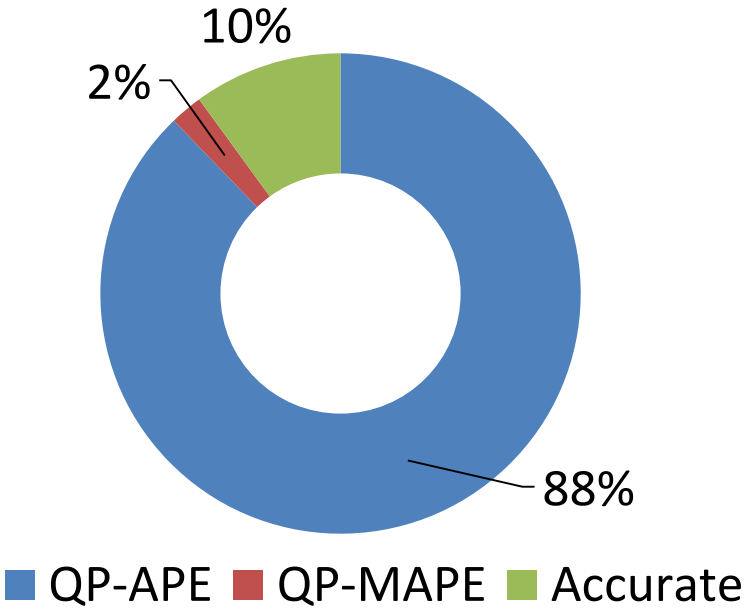
- **1.05-1.7X** savings for **NO** loss in output quality
- **1.18-2.1X** savings for **< 2.5%** quality loss
- **> 2.5X** savings for **< 7.5%** quality loss

Results Summary

QP-instructions in QUORA



Dynamic instruction count

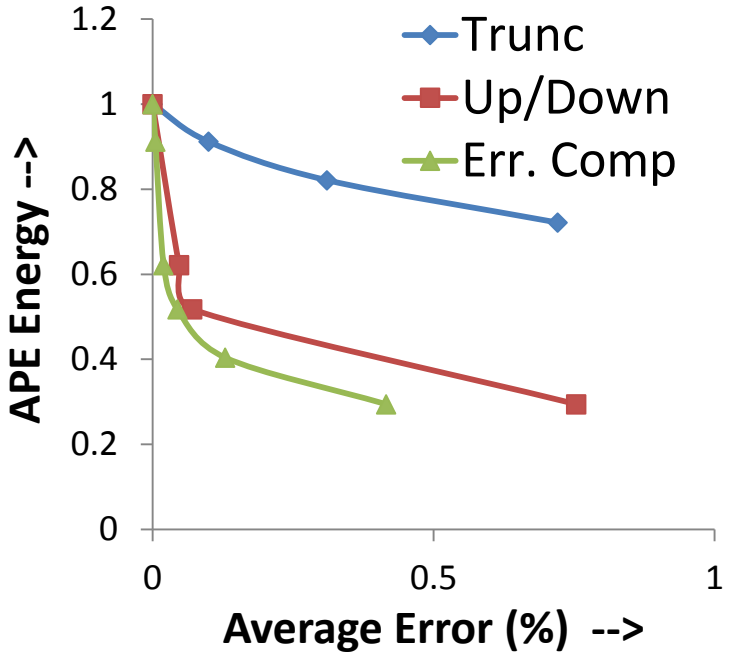


Energy

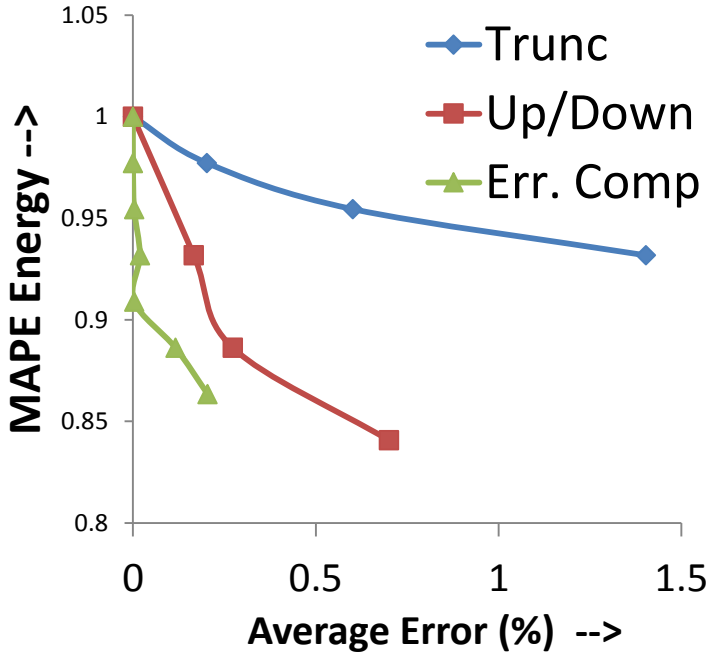
- **90%** of application energy in quality programmable instructions

Results Summary

Precision scaling mechanisms



MAC - APE



ACC - MAPE

- Precision scaling with error compensation provides superior energy vs. quality trade-off

Summary

- Intrinsic application resilience: A new dimension to optimize HW and SW
- Objective: Energy-efficient & programmable processor for approximate computing
- Quality programmable processors: Quality codified as part of the instruction set
- Quora: Quality programmable 1D/2D vector processor
  - Quality programmable ISA and microarchitecture